



TITLE:

文書作成で利用できる作図ツール
とKETpic (数学ソフトウェアと教育
: 数学ソフトウェアの効果的利用に
関する研究)

AUTHOR(S):

深澤, 謙次; 高遠, 節夫

CITATION:

深澤, 謙次 ...[et al]. 文書作成で利用できる作図ツールとKETpic (数学ソフトウェアと教育
: 数学ソフトウェアの効果的利用に関する研究). 数理解析研究所講究録 2012, 1780: 64-71

ISSUE DATE:

2012-03

URL:

<http://hdl.handle.net/2433/171832>

RIGHT:

文書作成で利用できる作図ツールと $\text{K}_\text{E}\text{T}_{\text{X}}$

呉工業高等専門学校・自然科学系分野 深澤 謙次 (Kenji Fukazawa)

Department of Natural Sciences,

Kure National College of Technology

東邦大学・薬学部 高遠 節夫 (Setsuo Takato)

Fucluty of Pharmaceutical Science,

Toho University

1 はじめに

数学や物理学の研究者や教育者の中には、論文の作成に $\text{L}_\text{A}\text{T}_{\text{E}}\text{X}$ を用いる者が多くいるが、教材の作成となると $\text{L}_\text{A}\text{T}_{\text{E}}\text{X}$ ではなく、Microsoft Word などのワープロを使用する者も、少なくない。その理由の 1 つは、 $\text{L}_\text{A}\text{T}_{\text{E}}\text{X}$ が図を扱うのが得意ではないことが考えられる。

教材にはきれいで正確な図が不可欠である。言葉や数式で説明してもなかなかわからないことが、図を 1 つ見せるだけで理解できることもある。したがって、 $\text{L}_\text{A}\text{T}_{\text{E}}\text{X}$ 文書にきれいで正確な図を簡単に入れられるようにならない限り、教材の作成に $\text{L}_\text{A}\text{T}_{\text{E}}\text{X}$ を使うようにはならない。

$\text{T}_\text{E}\text{X}$ 文書にきれいで正確な図を挿入するためのツールとして開発されたものの 1 つに $\text{K}_\text{E}\text{T}_{\text{X}}$ がある。 $\text{K}_\text{E}\text{T}_{\text{X}}$ は数学の配布用印刷教材の作成を目的に開発が始められ、数式処理システム (以下、CAS) 上で動作するパッケージとして提供されている。現在の開発は主に Scilab 上で行われている。

本論文では、他の作図ソフトウェアと比べて $\text{K}_\text{E}\text{T}_{\text{X}}$ がどれぐらい容易に利用できるかを考察する。作図の容易さを比較することが目的であるので、本論文で比較の参考とする図は単純なものを考える。比較する他の作図ソフトウェアとしては、Gnuplot, Asymptote, PGFplots を取り上げる。

ここでは簡単に $\text{K}_\text{E}\text{T}_{\text{X}}$ の紹介をする。 $\text{K}_\text{E}\text{T}_{\text{X}}$ の利用方法は以下のとおりである。

$\text{K}_\text{E}\text{T}_{\text{X}}$ では $\text{T}_\text{E}\text{X}$ 文書用の挿図を作成するために、 T_{pic} を利用する。 T_{pic} とは $\text{T}_\text{E}\text{X}$ 用に開発された図形プリプロセッサ及びそれが出力する special コマンドセットの名称である。 T_{pic} を用いて $\text{T}_\text{E}\text{X}$ 文書に図を挿入するには、図を描くための一連の T_{pic} のコマンドの並びをファイルに書き込み、そのファイルを $\backslash\text{input}$ 文を用いて $\text{T}_\text{E}\text{X}$ のマスターソースファイルに読み込めばよい。

$\text{K}_\text{E}\text{T}_{\text{X}}$ はこの T_{pic} のソースファイルを作成するための CAS 上で動作するプログラム群として実装されている。 $\text{K}_\text{E}\text{T}_{\text{X}}$ を用いることで、ユーザーは T_{pic} のコマンドを知らなくても T_{pic} を利用した図が作成できる訳である。この結果、 $\text{K}_\text{E}\text{T}_{\text{X}}$ には以下のような特徴が生まれている。

- $\text{T}_\text{E}\text{X}$ との親和性が良い (図の中に本文と同じ書体で数式が書ける)。

- 形と大きさに関して正確な図が描ける。
- 図の中に様々な装飾がつけられる。
- 豊かな表現力を持ったモノクロ線画が描ける。
- 修正が容易である。

K_εTpic を用いて挿図を作成する手順を模式的に図示すると、図1 のようになる。ユーザーは CAS 上で K_εTpic のコマンドを使って図を描くための一連のコマンドの並びを書き、 Tpic ファイルを作成する。このファイルを L^AT_EX ソースファイルに読み込みコンパイルすると、挿図入りの dvi ファイルが得られる。図を修正したい場合は、CAS 上にもどり K_εTpic のコマンドを修正後、同じことを繰り返す。

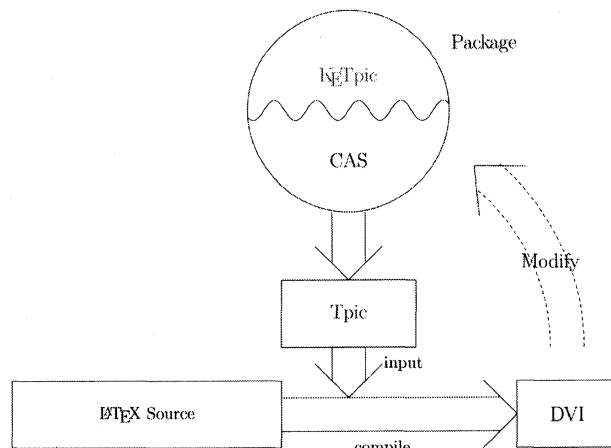


図1 K_εTpic による作図手順

コマンドリファレンスなどは以下のサイトから自由にダウンロードできる。

<http://ketpic.com>.

2 Gnuplot による作図例

Gnuplot は関数やデータの2次元もしくは3次元のグラフを作成するためのソフトウェアであり、研究者によく使われている。Gnuplot の特徴としては

- コマンドラインアプリケーションソフトウェアである
- 無料で使用できる
- 高機能である

などが上げられる。

Gnuplot で作成したグラフを L^AT_EX 文書に取り込む方法は、基本的には以下の2通りである。

1. グラフを eps 形式で保存し、`\includegraphics` コマンドで読み込む
2. グラフを L^AT_EX 形式で保存し、`\input` コマンドで読み込む

Gnuplot を使った簡単なグラフの例は 図2 のようになる (グラフを eps 形式で保存し, `\includegraphics` コマンドを利用). このグラフを描くための Gnuplot のコマンドは以下のようなシンプルなものである.

```
plot [-2*pi:2*pi][-4:4] 2*exp(-x**2/10), 3*sin(x)
set terminal postscript eps
set output "gp-simple-graph.eps"
```

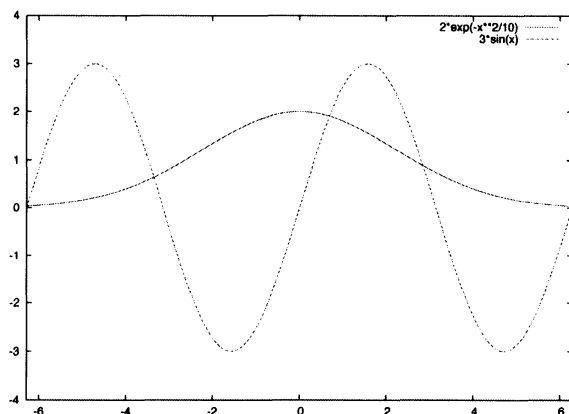


図2 Gnuplot による簡単なグラフの例

図を見ればわかるように, このグラフには座標軸がないなど, 数学教材用のグラフとしては適当ではない (物理教材としては問題ない). 図2 を数学教材に適切なものにする と図3 のようになる (グラフを \LaTeX 形式で保存し, `\input` コマンドを利用). このグラフを描くための Gnuplot のコマンドは以下のようなになる.

```
set border 0
set xzeroaxis
set yzeroaxis
set xtics axis
set ytics axis
set label "0" at -0.45, -0.3
set label "$x$" at 2*pi+0.1, 0 left
set label "$y$" at 0, 4.3 center
set xtics ("-$2\pi$" -2*pi, "$-\pi$" -pi, "$\pi$" pi, "$2\pi$" 2*pi)
set ytics ("-3" -3, "2" 2, "3" 3)
set label 1 "$y = 2 \exp\left(-\frac{x^2}{10}\right)$" at 3.5, 1 left
set label 2 "$y = 3 \sin x$" at -4, 3 left
unset key
plot [-2*pi:2*pi][-4:4] 2*exp(-x**2/10), 3*sin(x)
set terminal latex
set output "gp-simple-label.tex"
```

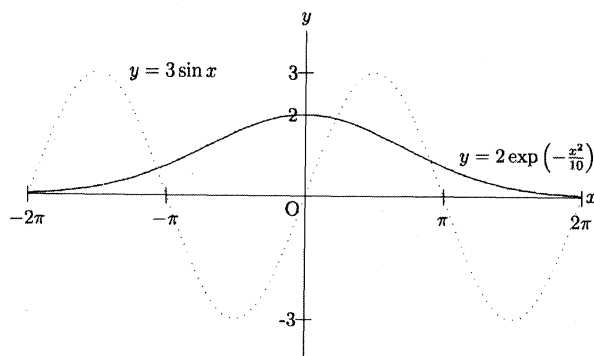


図3 Gnuplot による数学に適したグラフの例

図3を描くためのGnuplotのコマンドは、座標軸の描画やラベルの書き込みなどができるが、細かい微調整ができない（例えばグラフと“2”が重なってしまい、位置がずらせない）。また、図の一部に斜線を入れたり日本語を書き込むことは難しく、工夫がいる。

さらに、画面上では2つのグラフは色で区別され分かり易いが、グラフを \LaTeX 形式で保存し、`\input`コマンドで \LaTeX ファイルに読み込むと2つ目のグラフが点線で表示されてしまい識別しにくいことがわかる。このことから、Gnuplotは印刷教材の作成には向いていないと言える。

3 Asymptote による作図例

AsymptoteはMetaPostを発展させたベクトルグラフィック記述言語であり、以下のような特徴がある。

- ライセンスはGPLである
- UNIX, Linux, MacOS, Microsoft Windowsなどで動く
- 無料で使用できる
- \LaTeX に依存しないが、ラベルを \LaTeX 形式で書くことができる
- 高機能である

Asymptoteで作成したグラフを \LaTeX 文書に取り込む方法は、基本的には以下の2通りである。

1. Asymptoteコードを \LaTeX ソースファイルに直接書く
2. Asymptoteコードを別ファイルに書き、`\input`コマンドで読み込む

図3を描くためのAsymptoteのコマンドは以下ようになる。

```
\begin{asy}
  import graph;
  size(15cm, 0);
  real xmin=-2*pi, xmax=2*pi;
```

```

real ymin=-4, ymax=4;
real f(real x) {return 2*exp(-x^2/10);}
real g(real x) {return 3*sin(x);}
path p1 = graph(f, -2*pi, 2*pi, n=200);
path p2 = graph(g, -2*pi, 2*pi, n=200);
draw(p1, red);
draw(p2, blue);
xaxis(Label("$x$",position=EndPoint, align=NE),xmin=xmin,xmax=xmax,
      Ticks(scale(.7)*Label(align=E),NoZero,begin=true,beginlabel=true,
      end=true,endlabel=true,Step=2,
      Size=1mm,pTick=black,ptick=gray), Arrow);
yaxis(Label("$y$",position=EndPoint, align=NE),ymin=ymin,ymax=ymax,
      Ticks(scale(.7)*Label(),NoZero,begin=false,beginlabel=false,
      end=false,endlabel=false,Step=1,Size=1mm,size=.5mm,
      pTick=black,ptick=gray),Arrow);
label("0", (0,0), SW);
label("$y = 2 \exp\left(-\frac{x^2}{10}\right)$", (3.5,1), E);
label("$y = 3 \sin x$", (-4,3), E);
\end{asy}

```

Asymptote の場合、若干コマンドが手間が掛かることがわかる。このこととは別に、Asymptote を利用するためには \LaTeX のソースファイルに `asymptote` パッケージを読み込まなくてはならず、また、Asymptote を利用して空間図形を含んだ \LaTeX 文書を作成するには \TeX エンジンとして $\text{p}\text{\TeX}$ は使用できず、 $\text{pdf}\text{\TeX}$ を使用しなければならない。また、ソースファイルをコンパイルするためには（ファイル名を “sample.tex” とすると）

```

pdflatex sample
asy sample-1
pdflatex sample

```

としなければならない多少操作が煩わしい。

一方で、図の一部に斜線を入れるなどの様々な装飾を付け加えたり日本語を書き込んだりすることは難しくない。また、画面上でグラフを表示すると、2つのグラフは色で区別され分かり易い。しかし、Asymptote はモノクロ印刷して配布する文書を作成することを想定して設計されていないので、図によっては（特に空間図形など）分かり難いものになってしまう。したがって、印刷教材の作成に向いているとは言えない。

4 PGFplots による作図例

PGFplots は \TeX マクロパッケージ（Till Tantau の PGF パッケージを元にしてい）であり、 \TeX 文書に高品質の関数やデータのグラフ（2次元・3次元）を挿入する

ことができる。このため、 \TeX 文書での本文と図中でのフォントの一貫性を保つことができるという特徴がある。

PGFplots で作成したグラフを \LaTeX 文書に取り込む方法は、基本的には以下の 2 通りである。

1. PGFplots コードを \LaTeX ソースファイルに直接書く
2. PGFplots コードを別ファイルに書き、`\input` コマンドで読み込む

図 3 を描くための PGFplots のコマンドは以下のようなになる。

```
\begin{tikzpicture}
\begin{axis}[
axis x line=middle,
axis y line=middle,
xlabel={ $x$ },
ylabel={ $y$ },
xmin=-6.2831, xmax=6.2831,
ymin=-4, ymax=4,
width=0.8\textwidth
]
\addplot[mark=none,smooth]{2*exp(-x*x/10)};
\addplot[mark=none,smooth]{3*sin(deg(x))};
\end{axis}
\draw (4.1,3.4) node {$0$};
\draw (8.2,4.8) node {$y = 2 \exp\left(-\frac{x^2}{10}\right)$};
\draw node at (2.5,6.5) {$y = 3 \sin x$};
\end{tikzpicture}
```

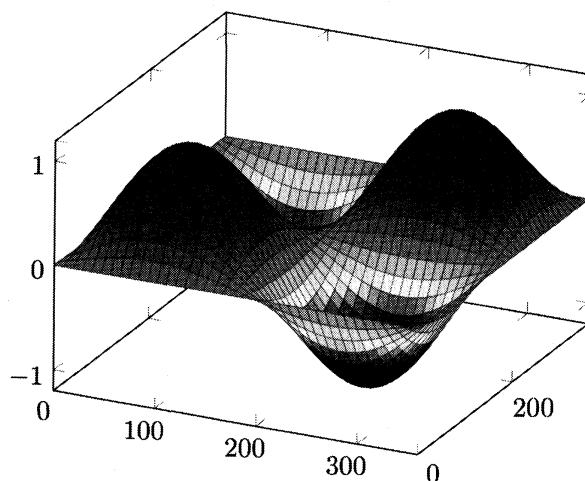


図 4 PGFplots による曲面の例

PGFplots の場合、座標軸などが比較的簡単なコマンドで描画できることがわかる。また、PGFplots を利用して \LaTeX 文書を作成するには、 \LaTeX のソースファイルに pgfplots パッケージを読み込まなくてはならず、Asymptote と同様、 \TeX エンジンとして $\text{p}\TeX$ は使用できず、 $\text{pdf}\TeX$ か $\text{X}\TeX$ を使用しなければならない。

一方で、図の一部に斜線を入れるなどの様々な装飾を付け加えたり日本語を書き込んだりすることは難しくない。また、画面上でグラフを表示すると、2つのグラフは色で区別され分かり易い。しかし、PGFplots はモノクロ印刷して配布する文書を作成することを想定して設計されていないので、図によっては（特に空間図形など）分かり難いものになってしまう（図4 参照）。したがって、印刷教材の作成に向いているとは言えない。

5 $\text{K}\epsilon\text{Tpic}$ による作図例

最後に $\text{K}\epsilon\text{Tpic}$ を利用した場合の例を示す。図3を描くための $\text{K}\epsilon\text{Tpic}$ のコマンドは以下ようになる（Maxima を利用した $\text{K}\epsilon\text{Tpic}$ のコマンド例）。

```
load("ketpic.mac")$
setwindow([-2*pi, 2*pi], [-4, 4])$
f: 2*exp(-x^2/10)$
g: 3*sin(x)$
pd1: plotdata(f, 30)$
pd2: plotdata(g, 200)$
openFile("maxima-simple-label.tex")$
beginpicture("1cm")$
drwline(pd1, pd2)$
htickmark(-2*pi, "-2\pi", -pi, "sw", "-\pi", pi, "sw", "\pi",
          2*pi, "se", "2\pi")$
vtickmark(-3, "-3", 2, "nw", "2", 3, "3")$
expr([4.0, 1], "e", "y = 2 \exp\left( -\frac{x^2}{10} \right)")$
expr([-4.0, 3], "e", "y = 3 \sin x")$
endpicture(1)$
closeFile()$
```

$\text{K}\epsilon\text{Tpic}$ の場合、座標軸などが簡単なコマンドで描画できることがわかる。 $\text{K}\epsilon\text{Tpic}$ を利用して \LaTeX 文書を作成する場合、 \TeX エンジンとして $\text{pdf}\TeX$ 、 $\text{X}\TeX$ だけでなく、 $\text{p}\TeX$ も使用できる。また、図の一部に斜線を入れるなどの様々な装飾を思い通りに付け加えたり日本語を書き込んだりすることが比較的簡単にできる。

また、図4と同じ曲面を $\text{K}\epsilon\text{Tpic}$ を用いて描くと図5のようになる。図4はカラーで描かれているので一見見やすい図のように思われるが、図4では2つある谷のうち、向こう側の谷の部分が手前の曲面に隠されて見えないのに対して、図5では隠れている部分が点線で表現されていることがわかる。また、図4はカラーで描かれているので教材

として大量に印刷して配布するのには向いていないと考えられるが、図5はモノクロの少ない線で描かれているので、モノクロ印刷に適していることがわかる。K_εTpic では、このような図が簡単に描けるように様々なコマンドが用意されている。

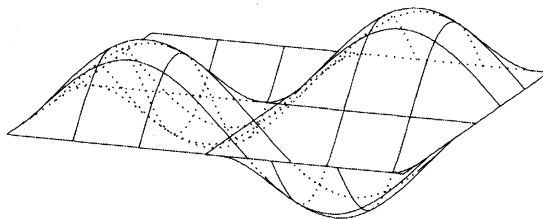


図5 K_εTpic for Scilab による曲面の例

6 Summary

本論文では、他の作図ソフトウェアと比べて K_εTpic がどれぐらい容易に利用できるかを、他の作図ソフトウェアとして Gnuplot, Asymptote, PGFplots と比較することによって考察した。作図の容易さを比較することが目的であるので、本論文で比較の参考とする図は単純なものを考えた。結果をまとめると以下ようになる。

- Gnuplot, Asymptote, PGFplots それぞれにグラフ自体は簡単に描ける。
- Gnuplot はグラフ以外の装飾（座標軸やラベルなど）を付けることは必ずしも簡単ではないが、Asymptote, PGFplots は日本語を書き込むことを含めて難しくはない。
- 空間図形も含めて考えると、Gnuplot, Asymptote, PGFplots は必ずしも印刷教材に向けた図を作成できる訳ではない。

数学向けの配布用印刷教材の作成に関しては、本論文で比較した作図ソフトウェアの中では、K_εTpic が最も容易であり適していると言える。

参考文献

- [1] CAST_εX 応用研究会編, K_εTpic で楽々T_εX グラフ, イーテキスト研究所, 2011.
- [2] A. Hammerlindl, J. Browman, and T. Prince, Asymptote: the Vector Graphics Language, 2004.
- [3] C. Feuersänger, Manual for Package PGFPLOTS, 2010.